1. Introduction

We are given a target area to fill with slate pieces. The pieces are given in a predetermined order. The slate pieces must be placed so that they won't overlap with the obstacles or with each other, and they must be connected with previously placed pieces. We need to cover the most area while using the minimum number of slate pieces. This problem is similar to the polyomino packing problem with additional constraints.

2. Solution

Because of the similarity between the puzzle and PPP, we decided to implement a backtracking algorithm to fill slate pieces on the target area. Because it runs in exponential time, we used a valuation function to decide which slate pieces should be used and where it should be placed in order to reach a good solution fast. We also maintain a tabu list to reduce the possibility of the algorithm stuck at local optima.

3. User interface

Our program has a GUI to visualize the placing process and to change various options for the algorithm.

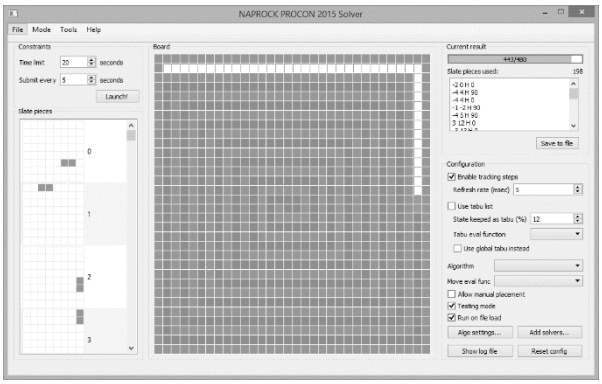

Figure 1The solver's GUI

4. Development environment

Programming languages: C++11 / Python (compiled with MSVC 2013 and GCC 4.9)

Additional frameworks/libraries: Qt

Platform supported: Windows, Linux